FieldServer Technologies

A Sierra Monitor Company

Driver Manual

(Supplement to the FieldServer Instruction Manual)

FS-8700-127 Hanning & Kahl

HCS-R

APPLICABILITY & EFFECTIVITY

Effective for all systems manufactured after April 2011

Driver Version:	1.00
Document Revision:	10

TABLE OF CONTENTS

1 HANNING & KAHL HCS-R DESCRIPTION

The Hanning & Kahl HCS-R driver allows the transfer of data from devices over RS-232 HCS-R protocol. Data is read from HCS-R and current data stored in a specified memory area. The driver can store the data with its loop data status distinct memory areas depending on wheter the data is on loop or from loop. The driver is based on section 10.5 of Hanning & Kahl - part number 40801121. The FieldServer can act as a Client or Server.

Max Nodes Supported

FieldServer Mode	Nodes	Comments
Client	1	Only 1 virtual client per port allowed on multidrop systems.
Server	1	This is a protocol limit, maximum devices that can be polled by Client

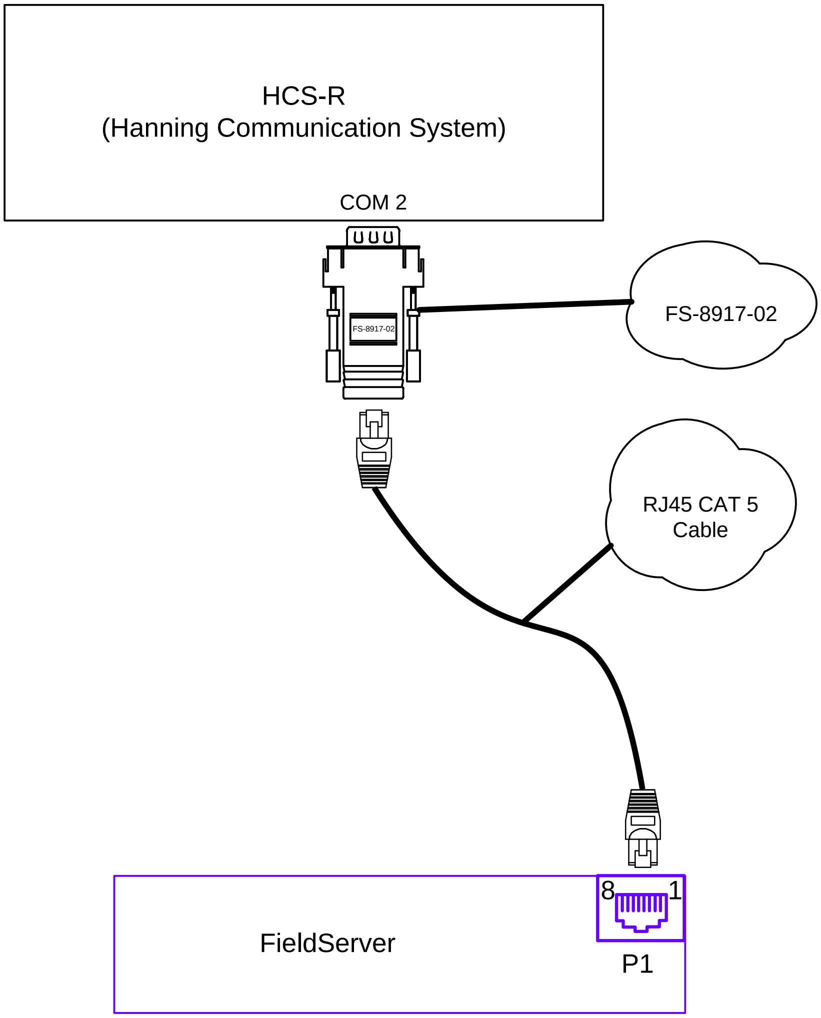
2 DRIVER SCOPE OF SUPPLY

2.1 Supplied by FieldServer Technologies for this driver

FieldServer Technologies PART #	Description
FS-8915-10	UTP cable (7 foot) for RS-232 use

3 HARDWARE CONNECTIONS

The FieldServer is connected to the Hanning & Kahl HCS-R as shown in connection drawing.
Configure the HCS-R according to manufacturer's instructions



FS-8917-02 Pinouts

FS Function	RJ45 Pin#	DB9F Pin#	Color
RX	1	3	White
GND	4	5	Green
TX	8	2	Blue

3.1 Hardware Connection Tips / Hints

- Loop back RTS CTS if it is required by the other device.

4 DATA ARRAY PARAMETERS

Data Arrays are “protocol neutral” data buffers for storage of data to be passed between protocols. It is necessary to declare the data format of each of the Data Arrays to facilitate correct storage of the relevant data.

Section Title
Data_Arrays

Column Title	Function	Legal Values
Data_Array_Name	Provide name for Data Array	Up to 15 alphanumeric characters
Data_Array_Format	Provide data format. Each Data Array can only take on one format.	Float, Bit, UInt16, SInt16, Byte.
Data_Array_Length	Number of Data Objects. Must be larger than the data storage area required by the Map Descriptors for the data being placed in this array.	1-10, 000

Example

```
//   Data Arrays
Data_Arrays
Data_Array_Name    , Data_Array_Format    , Data_Array_Length
DA_CURRENT         , Uint16               , 22
DI_FromLoop01      , Bit                  , 22
DI_OnLoop01        , Bit                  , 22
DI_FromLoop02      , Bit                  , 22
DI_OnLoop02        , Bit                  , 22
DI_FromLoop03      , Bit                  , 22
DI_OnLoop03        , Bit                  , 22
DI_FromLoop014     , Bit                  , 22
DI_OnLoop04        , Bit                  , 22
```

5 CONFIGURING THE FIELDSERVER AS A HCS-R CLIENT

For a detailed discussion on FieldServer configuration, please refer to the FieldServer Configuration Manual. The information that follows describes how to expand upon the factory defaults provided in the configuration files included with the FieldServer (See ".csv" sample files provided with the FieldServer).

This section documents and describes the parameters necessary for configuring the FieldServer to communicate with a HCS-R Server.

The configuration file tells the FieldServer about its interfaces, and the routing of data required. In order to enable the FieldServer for HCS-R communications, the driver independent FieldServer buffers need to be declared in the "Data Arrays" section, the destination device addresses need to be declared in the "Client Side Nodes" section, and the data required from the servers needs to be mapped in the "Client Side Map Descriptors" section. Details on how to do this can be found below.

Note that in the tables, * indicates an optional parameter, with the bold legal value being the default.

5.1 Client Side Connection Parametes

Section Title
Connections

Column Title	Function	Legal Values
Port	Specify which port the device is connected to the FieldServer	P1-P8[1]
Protocol	Specify protocol used	HCS-R HCSR H&K HCS-R
Baud*	Specify baud rate	**9600** (Vendor limitation)
Parity*	Specify parity	**None** (Vendor limitation)
Data_Bits*	Specify data bits	**8** (Vendor limitation)
Stop_Bits*	Specify stop bits	**1** (Vendor limitation)
Poll_Delay*	Time between internal polls	0-32000 seconds, **0.05 seconds**

Example

```
//   Client Side Connections

Connections
Port    , Protocol    , Baud    , Parity    , Poll_Delay
P1      , HCS-R       , 9600    , None      , 0.100s
```

[1] Not all ports shown are necessarily supported by the hardware. Consult the appropriate Instruction manual for details of the ports available on specific hardware.

5.2 Client Side Node Parameters

Section Title
Nodes

Column Title	Function	Legal Values
Node_Name	Provide name for Node	Up to 32 alphanumeric characters
Node_ID	HCS-R station address of physical server Node	240 (Protocol limit)
Protocol	Specify protocol used	HCS-R HCSR H&K HCS-R
Connection	Specify which port the device is connected to the FieldServer	P1-P8[2]

Example

```
//    Client Side Nodes

Nodes
Node_Name     , Node_ID     , Protocol     , Connection
PLC 1         , 240         , HCS-R        , P1
```

5.3 Client Side Map Descriptor Parameters

5.3.1 FieldServer Related Map Descriptor Parameters

Column Title	Function	Legal Values
Map_Descriptor_Name	Name of this Map Descriptor	Up to 32 alphanumeric characters
Data_Array_Name	Name of Data Array where data is to be stored in the FieldServer	0 to (Data_Array_Length-1) as specified in Section 4.
Data_Array_Offset	Starting location in Data Array	0 to (Data_Array_Length-1) as specified in Section 4.
Function	Function of Client Map Descriptor	RDBC, Passive_Client

[2] Not all ports shown are necessarily supported by the hardware. Consult the appropriate Instruction manual for details of the ports available on specific hardware.

5.3.2 Driver Specific Map Descriptor Parameters

Column Title	Function	Legal Values
Node_Name	Name of Node to fetch data from	One of the Node Names specified in Section 5.2.
HCSR_OnFrom	For Passive_Client Map Descriptor specify whether data From Loop or On Loop or both should be stored.	On, From, OnFrom
HCSR_Loop	For Passive_Client Map Descriptor specify loop number for which data should be stored.	1,2,3,4
Length*	Length of Map Descriptor 22 (min) for RDBC Map Descriptor: 24 if data_type is Dig_Input 21 for Passive_Client Map Descriptor. It also depends upon HCSR_Data_Stages parameter Refer to Appendix A.	22 or 24, **21**
Data_Type*	Specify how data will be stored. Refer to Appendix A.1	**Ana_Input** Digi_Input
HCSR_Data_Stages*	Specify the number of stages to keep history of data changes. 0 and 1 means no history	**0**,1,2,3

5.3.3 Timing Parameters

Column Title	Function	Legal Values
Scan_Interval	Rate at which data is polled	≥0.001s

5.4 Map Descriptor Example: Read Data

RDBC Map Descriptors are used to poll for data, Passive_Client Map Descriptors are used to store the data for various loops. Refer to Appendix A.1 for data storage format.

Function must be RDBC to enable polling.

The CMD_DI Map Descriptor polls HCS-R Node_A every 5 seconds.

```
Map_Descriptors
Map_Descriptor_Name   , Data_Array_Name   , Data_Array_Offset   , Function         , Node_name   , Length   , Scan_Interval
CMD_DI_HCSR_01        , DA_CURRENT        , 0                   , RDBC             , Node_A      , 22       , 5

Map_Descriptors
Map_Descriptor_Name   Data_Array_Name   Data_Array_Offset   Function         Node_name   Length   HCSR_Onfrom   HCSR_Loop
CMD_FromLoop_01       DI_FromLoop01     0                   Passive_Client   Node_A      21       From          1
CMD_OnLoop_01         DI_OnLoop01       0                   Passive_Client   Node_A      21       On            1
CMD_FromLoop_02       DI_FromLoop02     0                   Passive_Client   Node_A      21       From          2
CMD_OnLoop_02         DI_OnLoop02       0                   Passive_Client   Node_A      21       On            2
CMD_FromLoop_03       DI_FromLoop03     0                   Passive_Client   Node_A      21       From          3
CMD_OnLoop_03         DI_OnLoop03       0                   Passive_Client   Node_A      21       On            3
CMD_FromLoop_4        DI_FromLoop04     0                   Passive_Client   Node_A      21       From          4
CMD_OnLoop_04         DI_OnLoop04       0                   Passive_Client   Node_A      21       On            4
```

Data Array name defined in Section 4

Starting offset in Data Array to hold data

Passive_Client Map Descriptors are used to store data returned by response.

Maximum number of placeholders for data storage.

5.5 Map Descriptor Example: Read Data and Keep History

This Map Descriptor allows the storage of data history. The driver divides the Data Array into sections corresponding to the number of HCSR_Data_Stages. Current data (stage1) will be stored in the first section and oldest (last stage) data will be stored in the last section in the Data array. Data will be shifted to next Stage only on change.

```
Map_Descriptors
Map_Descriptor_Name   , Data_Array_Name   , Data_Array_Offset   , Function          , Node_Name   , Length   , Scan_Interval   , HCSR_Data_Stages
CMD_DI_HCSR_01        , DA_CURRENT        , 0                   , RDBC              , Node_A      , 72       , 5               , 3

Map_Descriptors
Map_Descriptor_Name   , Data_Array_Name   , Data_Array_Offset   , Function          , Node_Name   , Length   , HCSR_Onfrom     , HCSR_Loop   , HCSR_Data_Stages
CMD_FromLoop_01       , DI_FromLoop01     , 0                   , Passive_Client    Node_A        , 63       , From            , 1           , 3
CMD_OnLoop_01         , DI_OnLoop01       , 0                   , Passive_Client    Node_A        , 63       , On              , 1           , 3
CMD_FromLoop_02       , DI_FromLoop02     , 0                   , Passive_Client    Node_A        , 63       , From            , 2           , 3
CMD_OnLoop_02         , DI_OnLoop02       , 0                   , Passive_Client    Node_A        , 63       , On              , 2           , 3
CMD_FromLoop_03       , DI_FromLoop03     , 0                   , Passive_Client    Node_A        , 63       , From            , 3           , 3
CMD_OnLoop_03         , DI_OnLoop03       , 0                   , Passive_Client    Node_A        , 63       , On              , 3           , 3
CMD_FromLoop_4        , DI_FromLoop04     , 0                   , Passive_Client    Node_A        , 63       , From            , 4           , 3
CMD_OnLoop_04         , DI_OnLoop04       , 0                   , Passive_Client    Node_A        , 63       , On              , 4           , 3
```

The number of data stages (data changes) to be kept as history.

6 CONFIGURING THE FIELDSERVER AS A HCS-R SERVER

For a detailed discussion on FieldServer configuration, please refer to the FieldServer Configuration Manual. The information that follows describes how to expand upon the factory defaults provided in the configuration files included with the FieldServer (See ".csv" sample files provided with the FieldServer).

This section documents and describes the parameters necessary for configuring the FieldServer to communicate with a Hanning & Kahl HCS-R Client.

The configuration file tells the FieldServer about its interfaces, and the routing of data required. In order to enable the FieldServer for HCS-R communications, the driver independent FieldServer buffers need to be declared in the "Data Arrays" section, the FieldServer virtual Node(s) needs to be declared in the "Server Side Nodes" section, and the data to be provided to the clients needs to be mapped in the "Server Side Map Descriptors" section. Details on how to do this can be found below.

Note that in the tables, * indicates an optional parameter, with the bold legal value being the default.

6.1 Server Side Connection Parameters

Section Title		
Connections		
Column Title	**Function**	**Legal Values**
Port	Specify which port the device is connected to the FieldServer	P1-P8[3]
Protocol	Specify protocol used	HCS-R HCSR H&K HCS-R
Baud*	Specify baud rate	**9600** (Vendor Limitation)
Parity*	Specify parity	**None** (Vendor Limitation)
Data_Bits*	Specify data bits	**8** (Vendor Limitation)
Stop_Bits*	Specify stop bits	**1** (Vendor Limitation)
Server_Hold_Timeout*	Specifies the time that the FieldServer will reserve the Server side connection while waiting for the Client side to update data in Data_Array (if necessary)	>1.0s, **2.0s**

Example

```
//      Server Side Connections

Connections
Port    , Protocol    , Baud    , Parity
P8      , HCS-R       , 9600    , None
```

[3] Not all ports shown are necessarily supported by the hardware. Consult the appropriate Instruction manual for details of the ports available on specific hardware.

6.2 Server Side Node Parameters

Section Title
Nodes

Column Title	Function	Legal Values
Node_Name	Provide name for Node	Up to 32 alphanumeric characters
Node_ID	Station address of physical server Node	240
Protocol	Specify protocol used	HCS-R HCSR H&K HCS-R
Server_Hold_Timeout*	Specifies the time that the FieldServer will reserve the Server side connection while waiting for the Client side to update data in Data_Array (if necessary)	>1.0s, **2.0s**

Example

```
//    Server Side Nodes

Nodes
Node_Name    , Node_ID    , Protocol
HCSR_01      , 240        , HCS-R
```

6.3 Server Side Map Descriptor Parameters

6.3.1 FieldServer Related Map Descriptor Parameters

Column Title	Function	Legal Values
Map_Descriptor_Name	Name of this Map Descriptor	Up to 32 alphanumeric characters
Data_Array_Name	Name of Data Array where data is to be stored in the FieldServer	0 to (Data_Array_Length-1) as specified in Section 4.
Data_Array_Offset	Starting location in Data Array	0 to (Data_Array_Length-1) as specified in Section 4.
Function	Function of Server Map Descriptor	Passive
Server_Hold_Timeout*	Specifies the time that the FieldServer will reserve the Server side connection while waiting for the Client side to update data in Data_Array (if necessary)	>1.0s, **2.0s**

6.3.2 Driver Specific Map Descriptor Parameters

Column Title	Function	Legal Values
Node_Name	Name of Node to fetch data from	One of the Node names specified in Section 6.2
Length	Length of Map Descriptor	22
Data_Type*	Specify format of the data to Serve. Refer to Appendix A.2	Digi_Input **Ana_Input**

6.3.3 Map Descriptor Example. - SMD_HCSR_01 Map Descriptor to Serve HCS-R clients.

```
//  Server Side Map Descriptors

Map_Descriptors
Map_Descriptor_Name   , Data_Array_Name   , Data_Array_Offset   , Function   , Node_Name   , Length   , Scada_Hold_Timeout
SMD_HCSR_01           , DA_HCSR           , 0                   , Passive    , HCSR_01     , 22       , 5
```

Data Array name defined in Section 4

The starting location of data to serve within the Data Array

The number of items within the Data Array which will be used to prepare the response to the HCS-R Client

Appendix A. Reference

Appendix A.1. How Client Stores Data:

The HCS will be polled by an RDBC Map Descriptor and data will be stored in the DA_Current Data Array in the following fashion if parameter HCSR_Data_Stages >1:

- New data is stored as current data.
- If data is changed, last current data is moved to 2nd Stage, 2nd Stage data is moved to 3rd stage and 3rd stage data is lost.
- When a vehicle has HCS-R equipment installed, Byte 7 (Status of the loop) is equivalent to the "From Loop (0), On Loop (1) byte listed in Appendix A.1.1. The 20th offset will hold whether data is From Loop or On Loop. To achieve this, the FieldServer should be configured to store On-Loop and From-Loop data in the same Data Array. i.e. Map Descriptor Parameter HSCR_Onfrom should be set to Onfrom. Refer to Section 5.3.2

Appendix A.1. Data_Type is Ana_Input or not specified

Data Array Offset	Description
Current Data	
0 - 18	Data bits (0 - 1)
19	Receipt ok (0), Disturbance (1)
20	From Loop (0), On Loop (1)
21	Loop Number (1,2,3,4) as integer
2nd (Previous) Data	
24 – 42	Data bits (0 – 1)
43	Receipt ok (0), Disturbance (1)
44	From Loop (0), On Loop (1)
45	Loop Number (1,2,3,4) as integer
3rd (Prior to previous) Data	
48 – 66	Data bits (0 – 1)
67	Receipt ok (0), Disturbance (1)
68	From Loop (0), On Loop (1)
69	Loop Number (1,2,3,4) as integer

Appendix A.1.2. Data_Type is Digi_Input

Data Array Offset	Description
Current Data	
0 - 18	Data bits (0 - 1)
19	Receipt ok (0), Disturbance (1)
20	From Loop (0), On Loop (1)
21,22,23	3 bits for Loop Number Offsets 21 22 23 1 0 0 –Loop 1 0 1 0 –Loop 2 1 1 0 –Loop 3 0 0 1 –Loop 4
2nd (Previous) Data	
24 – 42	Data bits (0 – 1)
43	Receipt ok (0), Disturbance (1)
44	From Loop (0), On Loop (1)
45	Loop Number (1,2,3,4) in 3 bits as above
3rd (Prior to previous) Data	
48 – 66	Data bits (0 – 1)
67	Receipt ok (0), Disturbance (1)
68	From Loop (0), On Loop (1)
69	Loop Number (1,2,3,4) in 3 bits as above

The same data will be stored in a Loop based Data Array which will then have the last known data for each loop

Data Array Offset	Description
Current Data	
0 - 18	Data bits (0 – 1)
19	Receipt ok (0), Disturbance (1)
20	From Loop (0), On Loop (1)
2nd (Previous) Data	
21 – 39	Data bits (0 – 1)
40	Receipt ok (0), Disturbance (1)
41	From Loop (0), On Loop (1)
3rd (Prior to previous) Data	
42 – 60	Data bits (0 – 1)
61	Receipt ok (0), Disturbance (1)
62	From Loop (0), On Loop (1)

Appendix A.2. How Server expects Data:

The Server expects data in the following format in order to constitute its response to HCS-R Clients.

Appendix A.2.1. Data_Type is Ana_Input or not specified

Data Array Offset	Description
0 - 18	Data bits (0 - 1)
19	Receipt ok (0), Disturbance (1)
20	From Loop (0), On Loop (1)
21	Loop Number (0,1,2,3)

Appendix A.2.2. Data_Type is Digi_Input

Data Array Offset	Description
0 - 18	Data bits (0 - 1)
19	Receipt ok (0), Disturbance (1)
20	From Loop (0), On Loop (1)
21,22	2 bits for Loop Number Offsets 21 22 0 0 –Loop 1 1 0 –Loop 2 0 1 –Loop 3 1 1 –Loop 4